Chapter 1

Introduction to web development

Objectives (part 1)

Applied

1. Use a Chrome to run JavaScript applications that are on the Internet, your computer, or a local server.
2. Use Chrome's developer tools to find the JavaScript statement that caused an error in a JavaScript application.
3. Use an IDE or text editor such as VS Code to edit HTML, CSS, and JavaScript files.
4. If you're using an IDE or text editor such as VS Code that lets you run web applications from it, use your IDE or text editor to run an application.

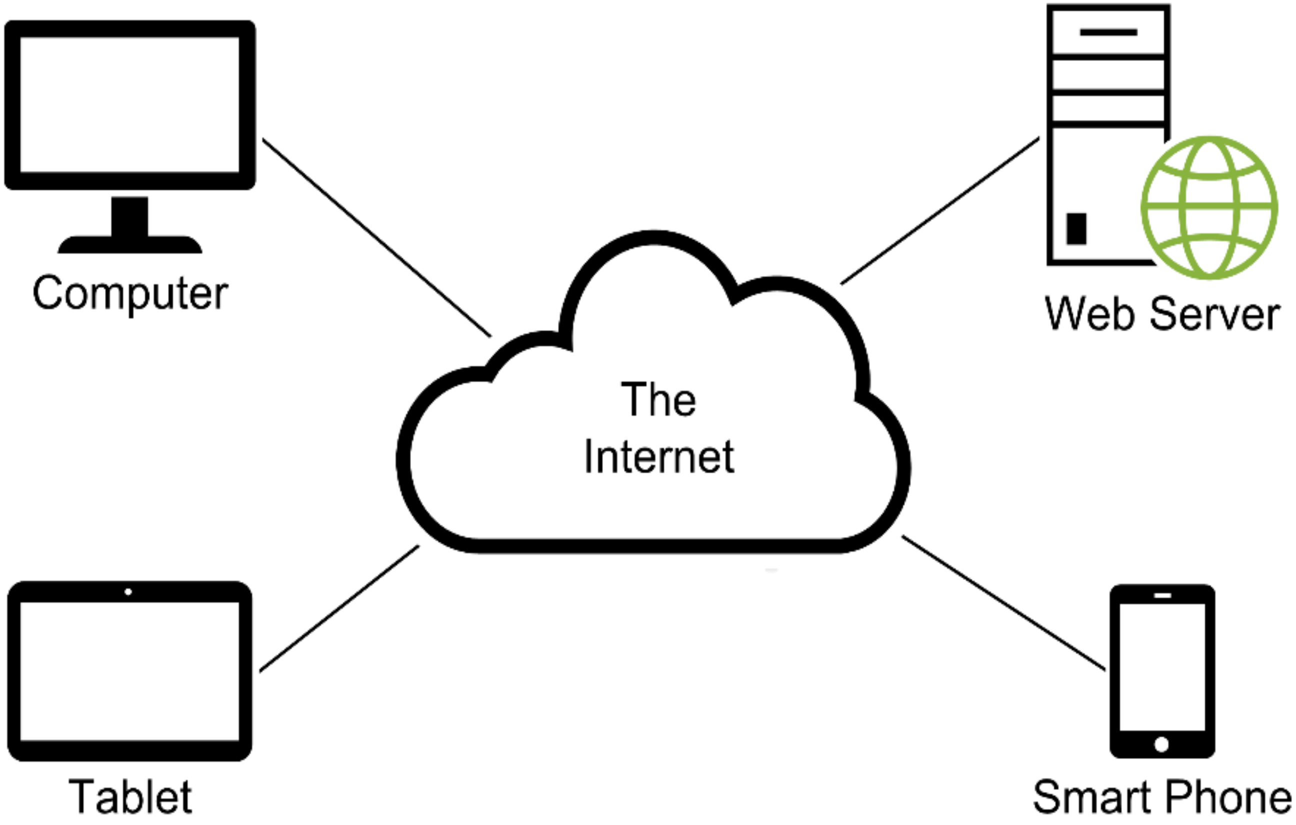
Knowledge

1. Describe the components of a web application.
2. Describe HTTP requests and responses.

Objectives (part 2)

3. Distinguish between static web pages and dynamic web pages.
4. Describe the use of JavaScript in a web application.
5. Describe the ECMAScript specification.
6. Describe browser support for the ECMAScript specification.
7. Describe the use of HTML and CSS.
8. Distinguish between the HTML5 semantic elements and the HTML div and span elements.
9. Describe the use of these HTML attributes: id, class, title, for, and name.
10. Describe the coding for these types of CSS selectors: type, id, and class.
11. Describe the components of a CSS style rule.
12. Describe the components of a URL.

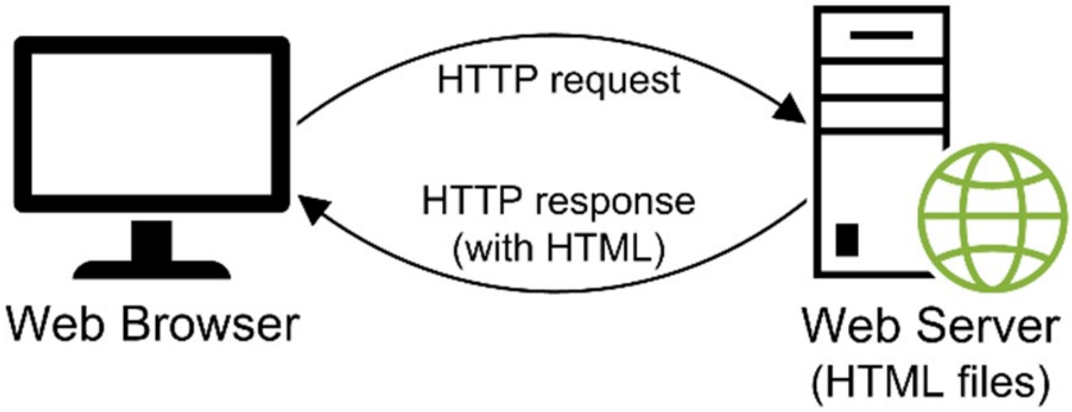
The components of a web application



Terms related to web applications

- client
- web browser
- web server
- network
- intranet
- local area network (LAN)
- Internet
- wide area network (WAN)
- Internet Service Provider (ISP)

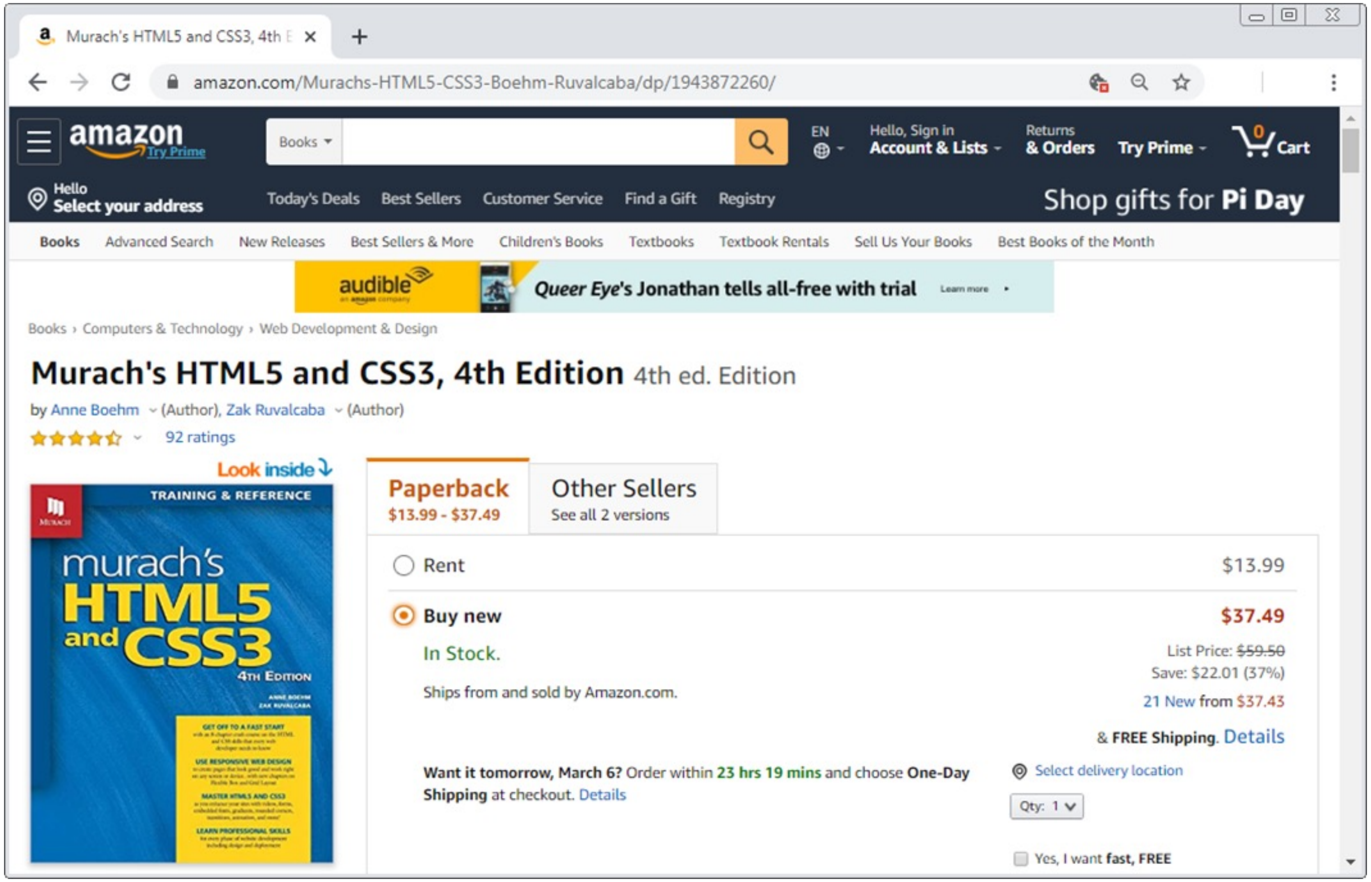
How a web server processes a static web page



Terms related to static web pages

- Hypertext Markup Language (HTML)
- static web page
- HTTP request
- HTTP response
- rendering a page

A dynamic web page at amazon.com



Murach's HTML5 and CSS3, 4th Edition 4th ed. Edition

by Anne Boehm (Author), Zak Ruvalcaba (Author)

92 ratings

○ Rent $13.99

● Buy new $37.49

In Stock.

List Price: ~~$59.50~~

Save: $22.01 (37%)

Ships from and sold by Amazon.com.

21 New from $37.43

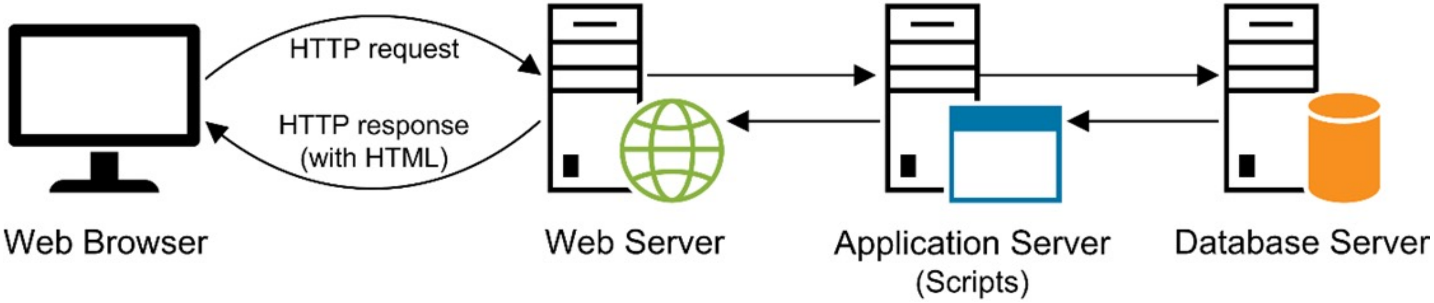
& FREE Shipping. Details

Want it tomorrow, March 6? Order within 23 hrs 19 mins and choose One-Day Shipping at checkout. Details

Select delivery location

□ Yes, I want fast, FREE

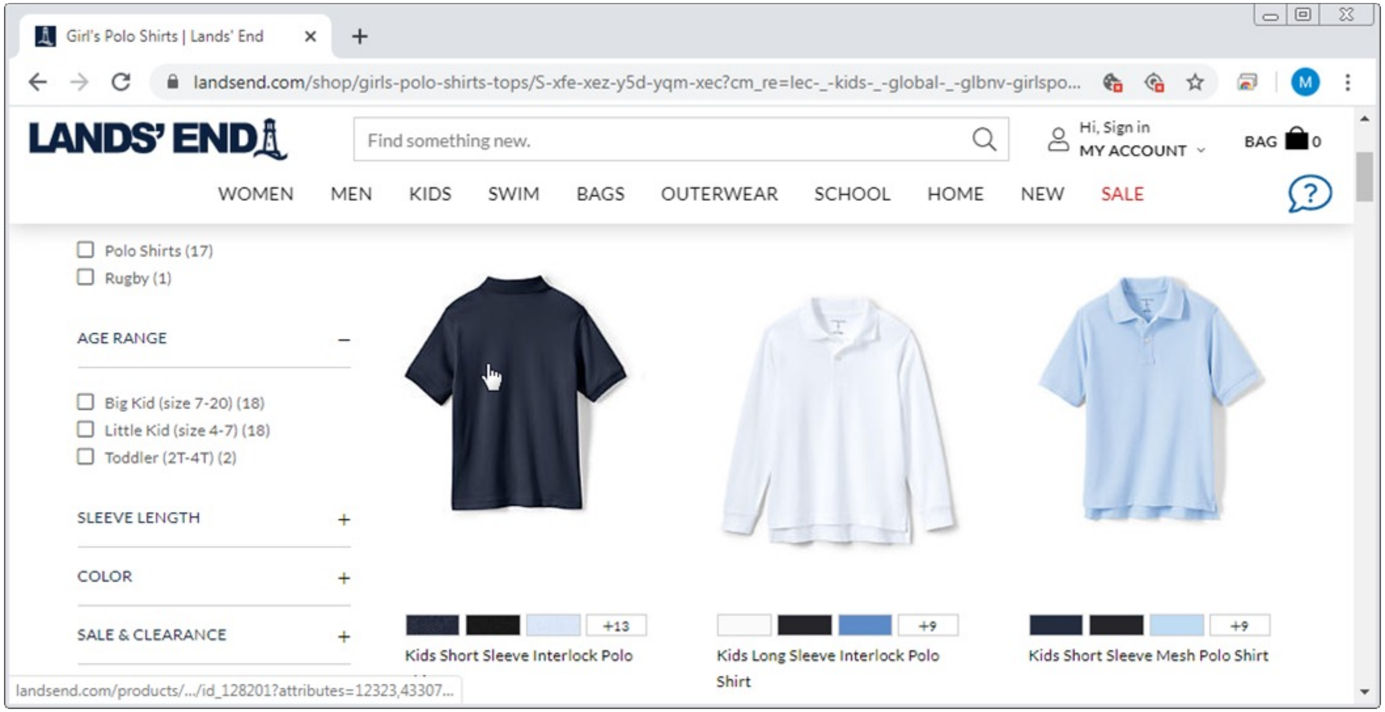
How a web server processes a dynamic web page



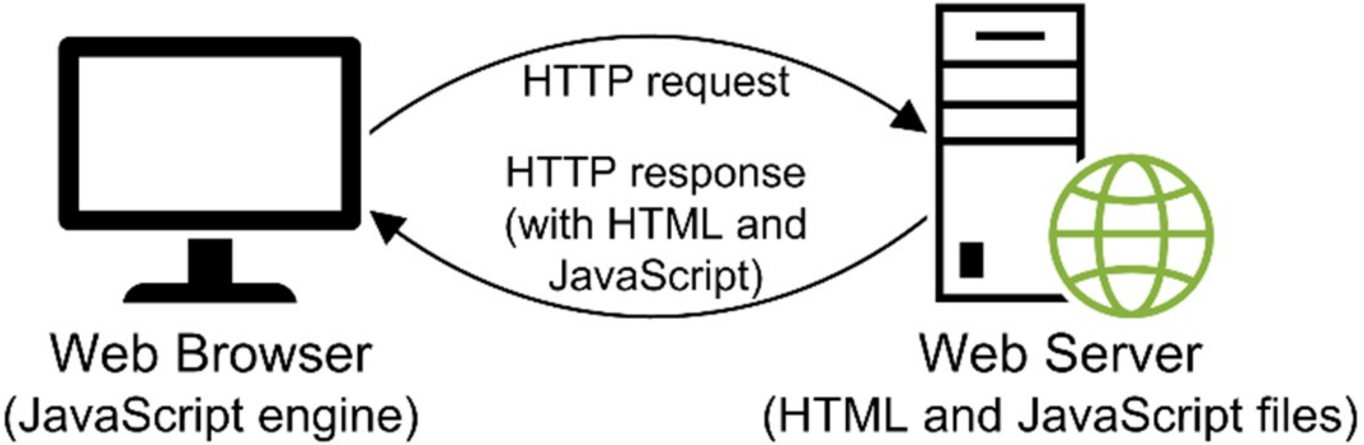
Terms related to dynamic web pages

- dynamic web page
- application server
- database server
- round trip
- server-side processing

A web page with image swaps and rollovers


Girl's Polo Shirts | Lands' End
landsend.com/shop/girls-polo-shirts-tops/S-xfe-xez-y5d-yqm-xec?cm_re=lec-_-kids-_-global-_-glbnv-girlspo...
LANDS' END
Find something new.
Hi, Sign in
MY ACCOUNT
BAG 0
WOMEN
MEN
KIDS
SWIM
BAGS
OUTERWEAR
SCHOOL
HOME
NEW
SALE
□ Polo Shirts (17)
□ Rugby (1)
AGE RANGE –
□ Big Kid (size 7-20) (18)
□ Little Kid (size 4-7) (18)
□ Toddler (2T-4T) (2)
SLEEVE LENGTH +
COLOR +
SALE & CLEARANCE +
+13
Kids Short Sleeve Interlock Polo
+9
Kids Long Sleeve Interlock Polo Shirt
+9
Kids Short Sleeve Mesh Polo Shirt
landsend.com/products/.../id_128201?attributes=12323,43307...

How JavaScript fits into this architecture


HTTP request
HTTP response (with HTML and JavaScript)
Web Browser
(JavaScript engine)
Web Server
(HTML and JavaScript files)

Terms related to client-side processing

- scripting language
- JavaScript engine
- jQuery
- client-side processing

Three of the many uses of JavaScript and jQuery

- Data validation
- Image swaps and rollovers
- Slide shows

The versions and release dates of the ECMAScript specification

Version	Release date
`1`	June 1997
`2`	June 1998
`3`	December 1999
`4`	Abandoned (never released)
`5`	December 2009
`5.1`	June 2011
`2015`	June 2015
`2016`	June 2016
`2017`	June 2017
`2018`	June 2018
`2019`	June 2019
`2020`	June 2020

Some additions in recent versions (part 1)

ES5

- Allows you to run in strict mode.
- Adds several methods that make it easier to work with arrays and objects.
- Adds a built-in way to work with JavaScript Object Notation (JSON).

ES2015 (ES6)

- Adds several syntactic improvements that make code easier to read and understand.
- Adds block scope and easier ways to work with classes.
- Adds arrow functions, iterators, and Promises for working with asynchronous code.

Some additions in recent versions (part 2)

ES2016

- Adds a simpler syntax for computation with powers.
- Adds a method to check if an array includes a specified element.

ES2017

- Adds async functions and the await keyword for working with Promises.

ES2018

- Adds asynchronous iteration.
- Adds more regular expression features.

Some additions in recent versions (part 3)

ES2019

- Adds new string, array, and object methods.
- Adds improvements to the JSON object.

ES2020

- Adds a new BigInt data type.
- Adds new operators for dealing with nulls.

The browsers that support ECMAScript

Browser	Version
Chrome	79 and above
Edge	79 and above
Firefox	68 and above
Opera	66 and above
Safari	12.1 and above

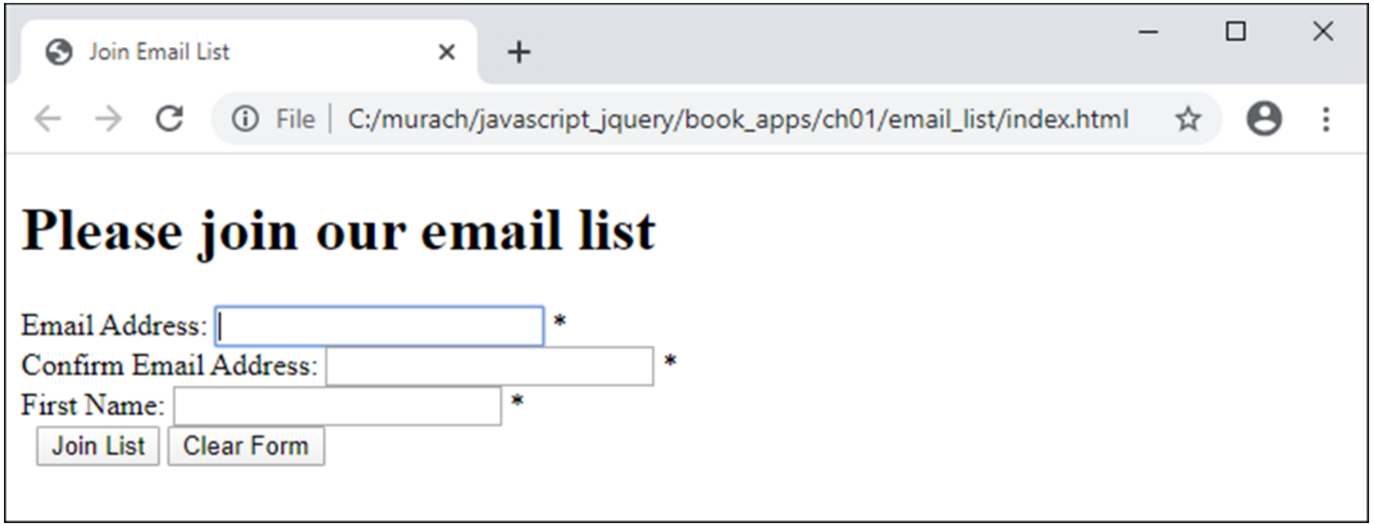
The URL for a browser compatibility table

http://kangax.github.io/compat-table/

Note

- At the time of this writing, the features of ECMAScript through ES2019 are supported by all the modern browsers listed here.

An HTML file (index.html) in a browser with no CSS applied to it



Join Email List
File
C:/murach/javascript_jquery/book_apps/ch01/email_list/index.html

Please join our email list

Email Address: *
Confirm Email Address: *
First Name: *

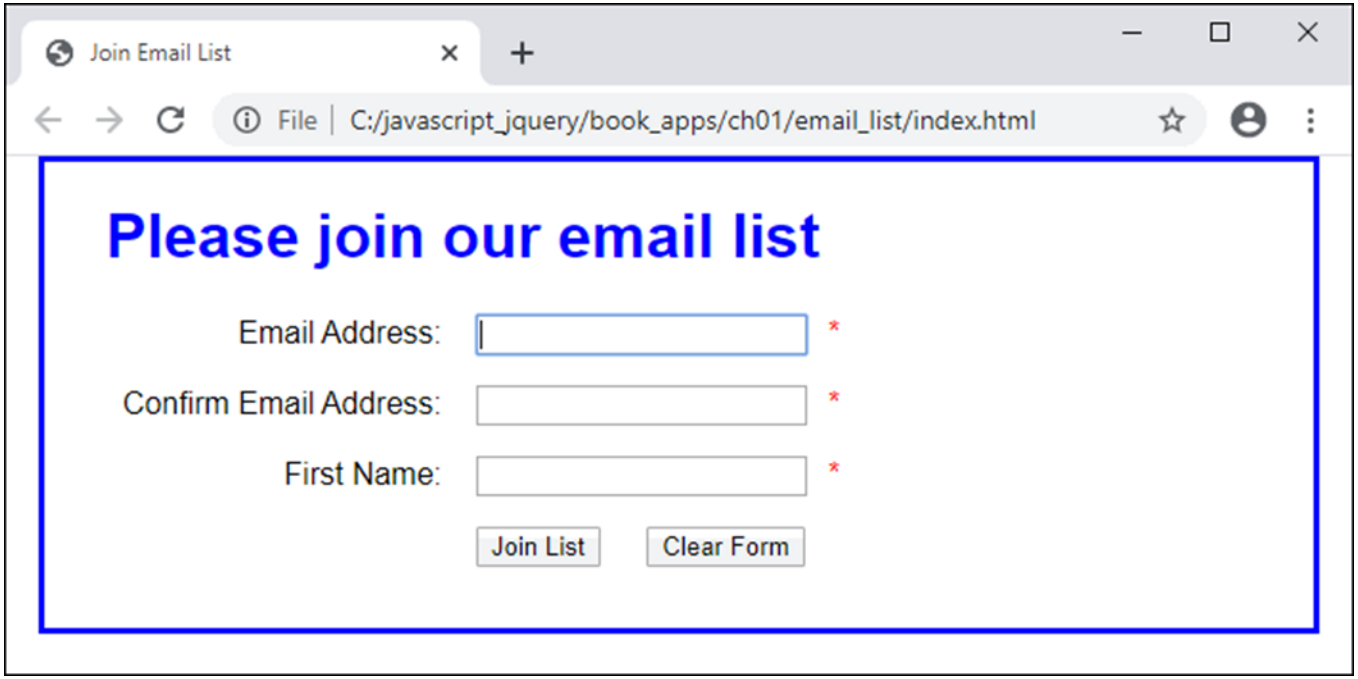
The HTML file named index.html (part 1)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport"
        content="width=device-width, initial-scale=1">
    <title>Join Email List</title>
    <!-- link and style elements go here -->
</head>
<body>
    <main>
        <h1>Please join our email list</h1>
        <form id="email_form" action="join.html"
              method="get">
            <div>
                <label for="email_1">Email Address:
                </label>
                <input type="text" id="email_1"
                       name="email_1">
                <span id="email_1_error">*</span>
            </div>
```

The HTML file named index.html (part 2)

```
            <div>
                <label for="email_2">
                    Re-enter Email Address:</label>
                <input type="text" id="email_2" name="email_2">
                <span id="email_2_error">*</span>
            </div>
            <div>
                <label for="first_name">First Name</label>
                <input type="text" id="first_name"
                       name="first_name">
                <span id="first_name_error">*</span>
            </div>
            <div>
                <label>&nbsp;</label>
                <input type="submit" id="join_list"
                       value="Join List">
                <input type="button" id="clear_form"
                       value="Clear Form">
            </div>
        </form>
    </main>
    <!-- script elements go here -->
</body>
</html>
```

The web page in a browser after CSS has been applied to it



Please join our email list

Email Address: *

Confirm Email Address: *

First Name: *

The link element in the HTML head element that applies the CSS file

```
<link rel="stylesheet" href="email_list.css">
```

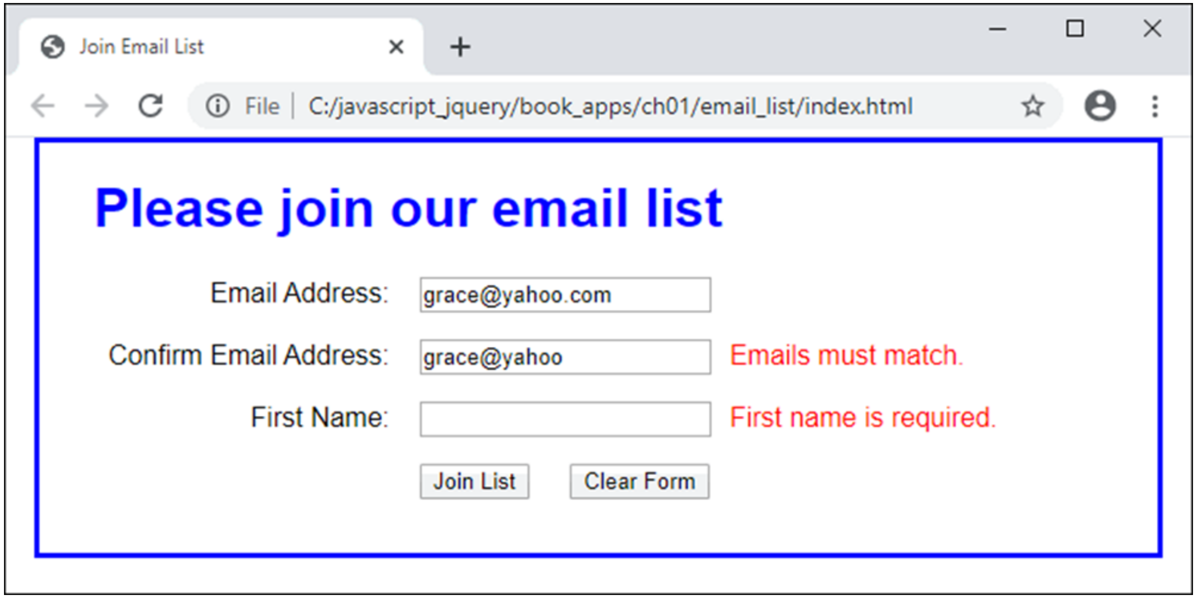
The CSS file named email_list.css (part 1)

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: white;
    margin: 0 auto;
    width: 670px;
    border: 3px solid blue;
    padding: 0 2em 1em;
}
h1 {
    color: blue;
}
```

The CSS file named email_list.css (part 2)

```
 div {
     margin-bottom: 1em;
 }
 label {
     display: inline-block;
     width: 11em;
     text-align: right;
 }
 input {
     margin-left: 1em;
     margin-right: 0.5em;
 }
 span {
     color: red;
}
```

The web page in a browser with JavaScript used for data validation



Please join our email list

Email Address: grace@yahoo.com

Confirm Email Address: grace@yahoo Emails must match.

First Name: First name is required.

The script element in the HTML body element that adds the JavaScript file

```
<script src="email_list.js"></script>
```

The email_list.js file (part 1)

```
const $ = selector => document.querySelector(selector);

const joinList = evt => {
    // get user entries from text boxes
    const email1 = $("#email_1").value;
    const email2 = $("#email_2").value;
    const firstName = $("#first_name").value;

    // check user entries
    let isValid = true;
    if (email1 == "") {
        $("#email_1_error").textContent = "Email is required.";
        isValid = false;
    } else {
        $("#email_1_error").textContent = "";
    }
    if (email1 != email2) {
        $("#email_2_error").textContent = "Emails must match.";
        isValid = false;
    } else {
        $("#email_2_error").textContent = "";
    }
```

The email_list.js file (part 2)

```
    if (firstName == "") {
        $("#first_name_error").textContent =
            "First name is required.";
        isValid = false;
    } else {
        $("#first_name_error").textContent = "";
    }

    // cancel form submit if any user entries are invalid
    if ( !isValid ) {
        evt.preventDefault();
    }
};
```

The email_list.js file (part 3)

```
const clearForm = () => {
    // clear text boxes
    $("#email_1").value = "";
    $("#email_2").value = "";
    $("#first_name").value = "";

    // clear span elements
    $("#email_1_error").textContent = "*";
    $("#email_2_error").textContent = "*";
    $("#first_name_error").textContent = "*";

    // set focus on first text box after resetting the form
    $("#email_1").focus();
};

document.addEventListener("DOMContentLoaded", () => {
    // hook up click events for both buttons
    $("#join_list").addEventListener("click", joinList);
    $("#clear_form").addEventListener("click", clearForm);

    // set focus on first text box after the form loads
    $("#email_1").focus();
});
```

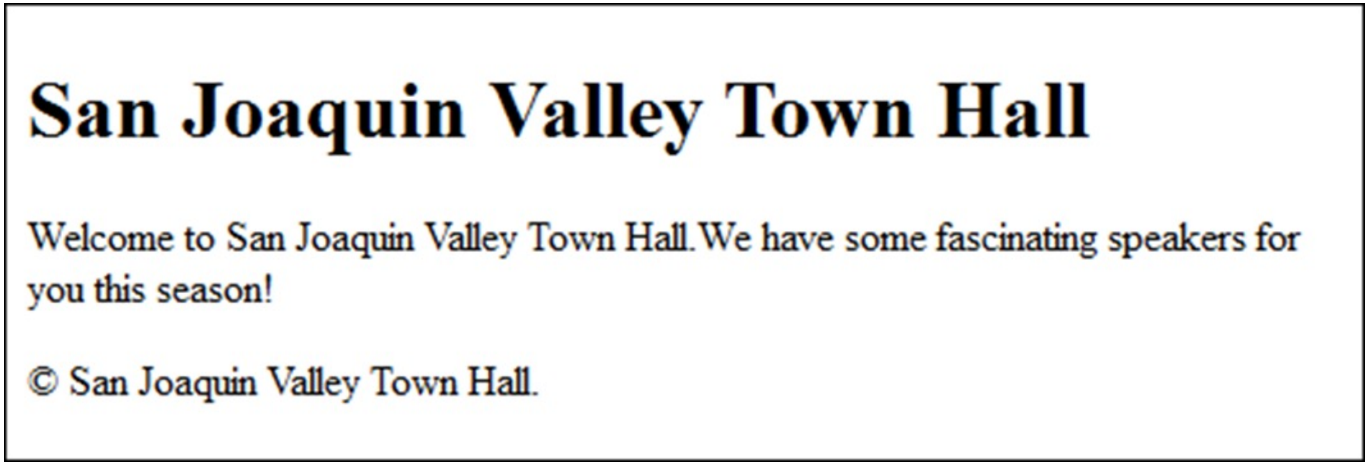
The primary HTML5 semantic elements

```
header
main
section
article
aside
nav
figure
footer
```

A page that's structured with HTML5 elements

```
<body>
    <header>
        <h1>San Joaquin Valley Town Hall</h1>
    </header>
    <main>
        <p>Welcome to San Joaquin Valley Town Hall.
           We have some fascinating speakers for you this
           season!</p>
    </main>
    <footer>
        <p>&copy; San Joaquin Valley Town Hall.</p>
    </footer>
</body>
```

The page displayed in a web browser



San Joaquin Valley Town Hall

Welcome to San Joaquin Valley Town Hall.We have some fascinating speakers for you this season!

© San Joaquin Valley Town Hall.

The CDN for the JavaScript shiv for HTML5 compatibility

```
http://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.js
```

The div and span elements

Element	Description
`div`	A block element that provides a container for other elements.
`span`	An inline element that lets you identify text that can be formatted with CSS.

HTML div elements for a JavaScript application

```
 <section id="faqs">
     <h1>jQuery FAQs</h1>
     <h2>What is JavaScript?</h2>
     <div>
         // contents
     </div>
     <h2>What is jQuery?</h2>
     <div>
         // contents
     </div>
     <h2>Why is jQuery becoming so popular?</h2>
     <div>
         // contents
     </div>
</section>
```

HTML span elements for a JavaScript application

```
 <div>
     <label for="email_1">Email Address:</label>
     <input type="text" id="email_1" name="email_1">
     <span id="email_1_error">*</span>
 </div>
 <div>
     <label for="email_2">Re-enter Email Address:</label>
     <input type="text" id="email_2" name="email_2">
     <span id="email_2_error">*</span>
 </div>
 <div>
     <label for="first_name">First Name</label>
     <input type="text" id="first_name“
            name="first_name">
     <span id="first_name_error">*</span>
</div>
```

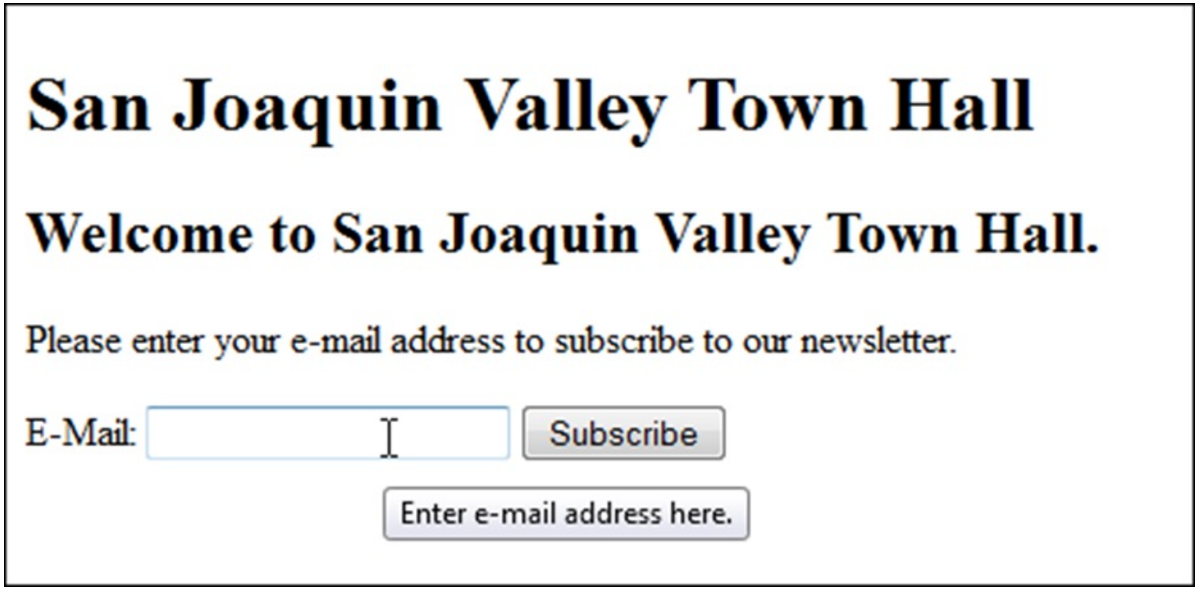
The basic HTML attributes

```
id
class
name
for
title
```

HTML that uses these attributes

```
 <body>
     <h1>San Joaquin Valley Town Hall</h1>
     <h2 class="first_h2">Welcome to San Joaquin Valley
         Town Hall.</h2>
     <p>Please enter your e-mail address to subscribe to
        our newsletter.</p>
     <form id="email_form" name="email_form"
           action="join.html" method="get">
         <label for="email">E-Mail: </label>
         <input type="text" id="email" name="email"
                title="Enter e-mail address here.">
         <input type="button" value="Subscribe">
     </form>
</body>
```

The HTML in a web browser with a tooltip displayed for the text box



San Joaquin Valley Town Hall

Welcome to San Joaquin Valley Town Hall.

Please enter your e-mail address to subscribe to our newsletter.

E-Mail:

Two ways to provide styles

Use an external style sheet by coding a link element in the head section

```
<link rel="stylesheet" href="styles/main.css">
```

Embed the styles in the head section

```
<style>
    body {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 87.5%; }
    h1 { font-size: 250%; }
</style>
```

The sequence in which styles are applied

- Styles from an external style sheet
- Embedded styles

A head element that includes two style sheets

```
<head>
    <title>San Joaquin Valley Town Hall</title>
    <link rel="stylesheet" href="../styles/main.css">
    <link rel="stylesheet" href="../styles/speaker.css">
</head>
```

The sequence in which styles are applied

- From the first external style sheet to the last

HTML that can be selected by type, id, or class

```
<body>
    <main>
        <h1>The Speaker Lineup</h1>
        <p class="blue">October 19: Jeffrey Toobin</p>
        <p class="blue">November 16: Andrew Ross Sorkin</p>
    </main>
    <footer>
        <p id="copyright" class="blue right">
           Copyright SJV Town Hall</p>
    </footer>
</body>
```

CSS style rules that select by type, id, and class

Elements by type

```
body {
    font-family: Arial, Helvetica, sans-serif;
    width: 400px;
    margin: 1em auto; }
main {
    display: block;
    padding: 1em;
    border: 2px solid black; }
h1 { margin: 0 0 .25em; }
p { margin: .25em 0 .25em 3em; }
```

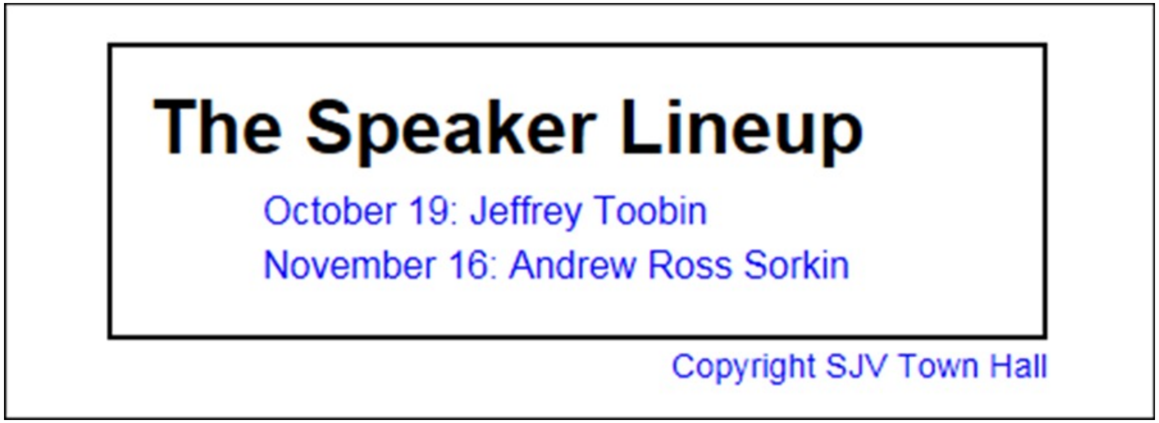
One element by ID

```
#copyright { font-size: 90%; }
```

Elements by class

```
.blue { color: blue; }
.right { text-align: right; }
```

The HTML elements displayed in a browser



The Speaker Lineup

October 19: Jeffrey Toobin

November 16: Andrew Ross Sorkin

Copyright SJV Town Hall

The CSS file for a typical application (part 1)

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: white;
    margin: 0 auto;
    width: 670px;
    border: 3px solid blue;
    padding: 0 2em 1em;
}
h1 {
    color: blue;
}
div {
    margin-bottom: 1em;
}
label {
    display: inline-block;
    width: 11em;
    text-align: right;
}
```

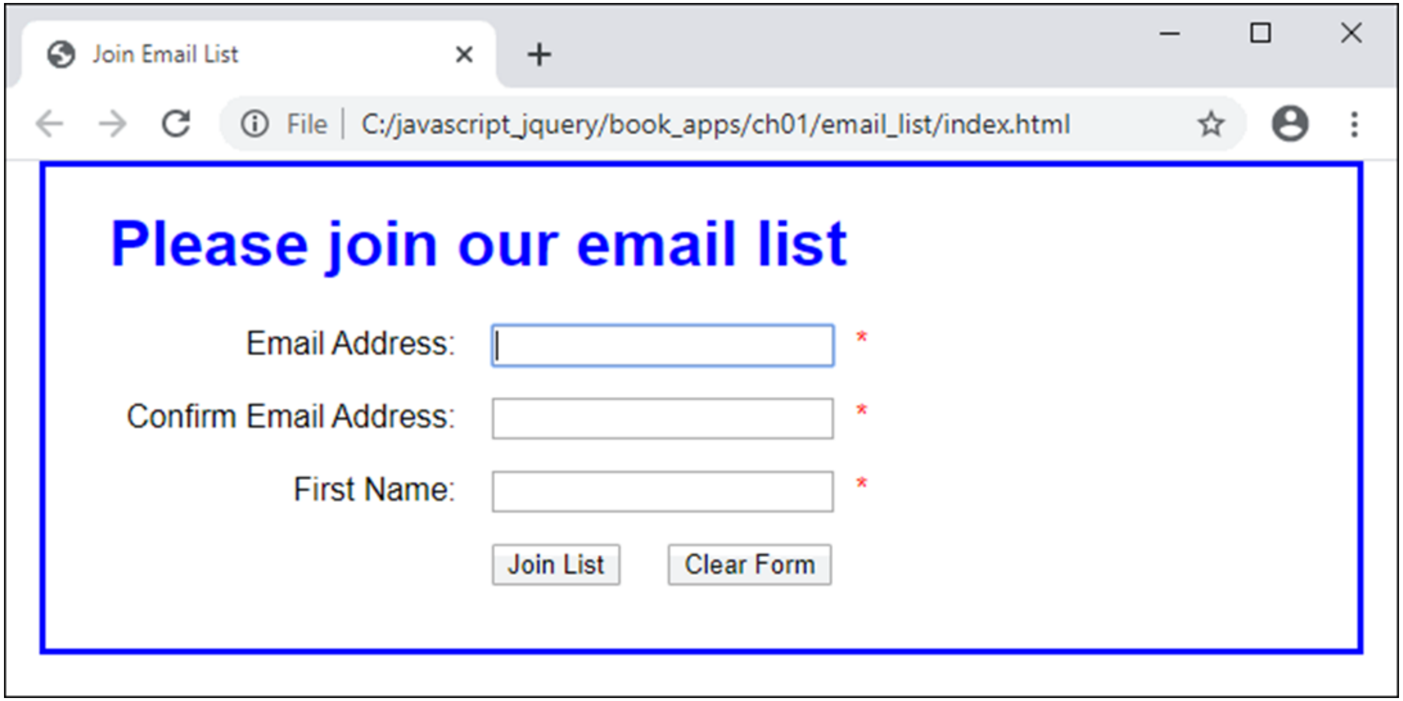
The CSS file for a typical application (part 2)

```
input {
    margin-left: 1em;
    margin-right: 0.5em;
}
span {
    color: red;
}
```

Terms related to CSS

- Cascading Style Sheets (CSS)
- style rule
- property declaration
- property name
- property value

The web page at c:/javascript_jquery/book_apps/ch01/email_list/index.html



Please join our email list

Email Address:		*
Confirm Email Address:		*
First Name:		*

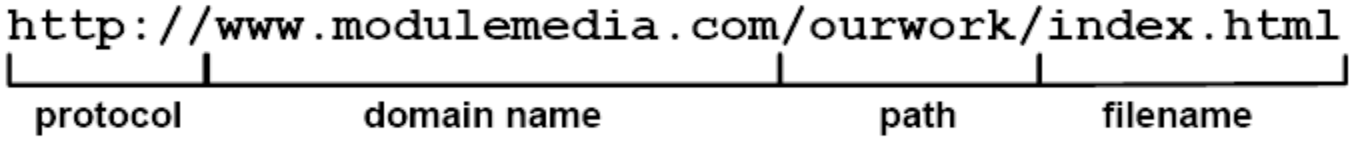
Four ways to run an HTML page that's on your own server or computer

- From your browser, use the Ctrl+O shortcut key combination to start the Open command. Then, browse to the HTML file and double-click on it.
- Use File Explorer (Windows) or Finder (macOS) to find the HTML file, and double-click on it.
- Use the features of your text editor or IDE.
- Click on a link in the current web page to load the next web page.

Two ways to run an HTML page on the Internet

- Enter the URL of the web page into the browser’s address bar.
- Click on a link in the current web page to load the next web page.

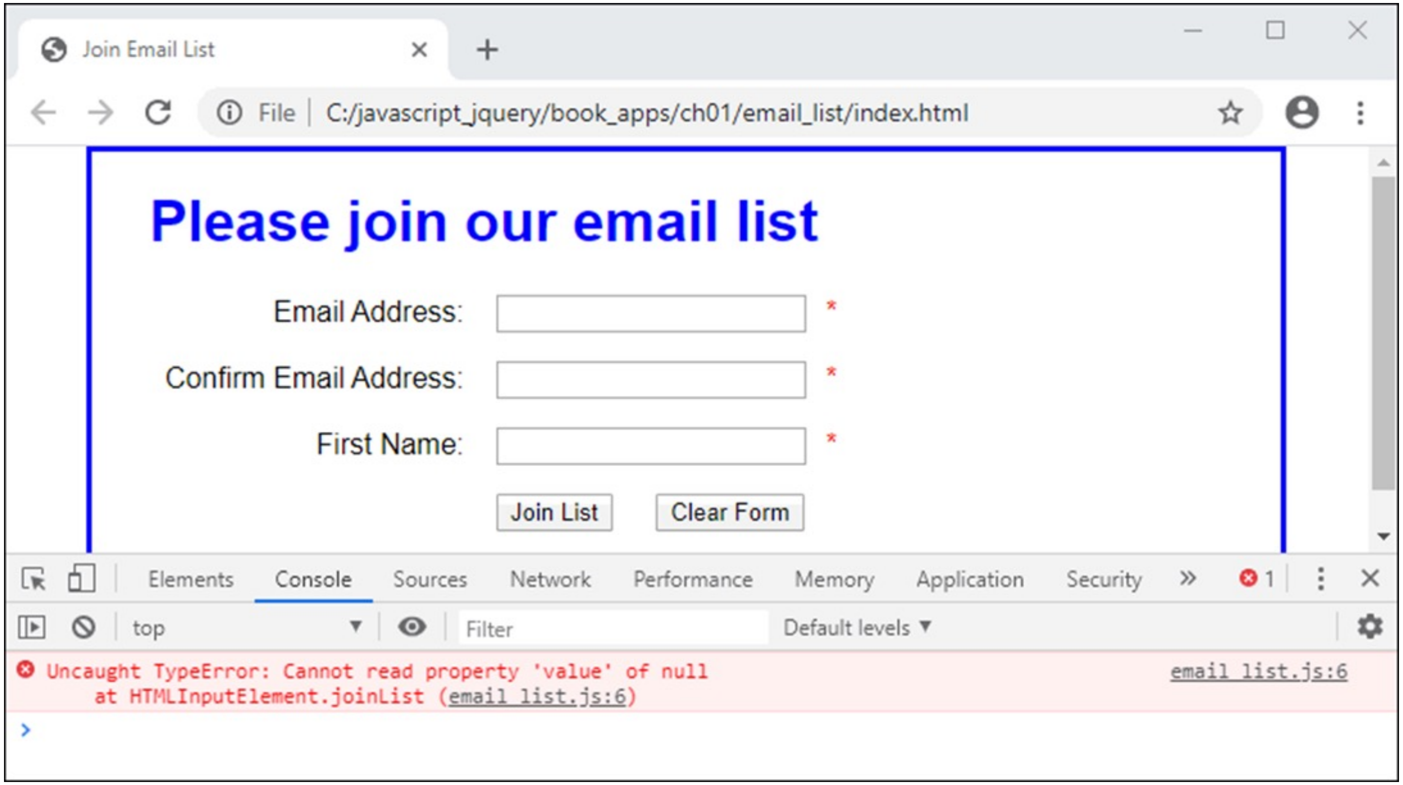
The components of an HTTP URL on the Internet



What happens if you omit parts of a URL

- If you omit the protocol, the default of http:// or https:// will be used.
- If you omit the filename, the default document name for the web server will be used.
- The default document name is typically index.html, default.htm, or some variation.

Chrome with an open Console panel that shows an error



Please join our email list

Email Address: *

Confirm Email Address: *

First Name: *

Join List   Clear Form

```
Uncaught TypeError: Cannot read property 'value' of null          email_list.js:6
    at HTMLInputElement.joinList (email_list.js:6)
>
```

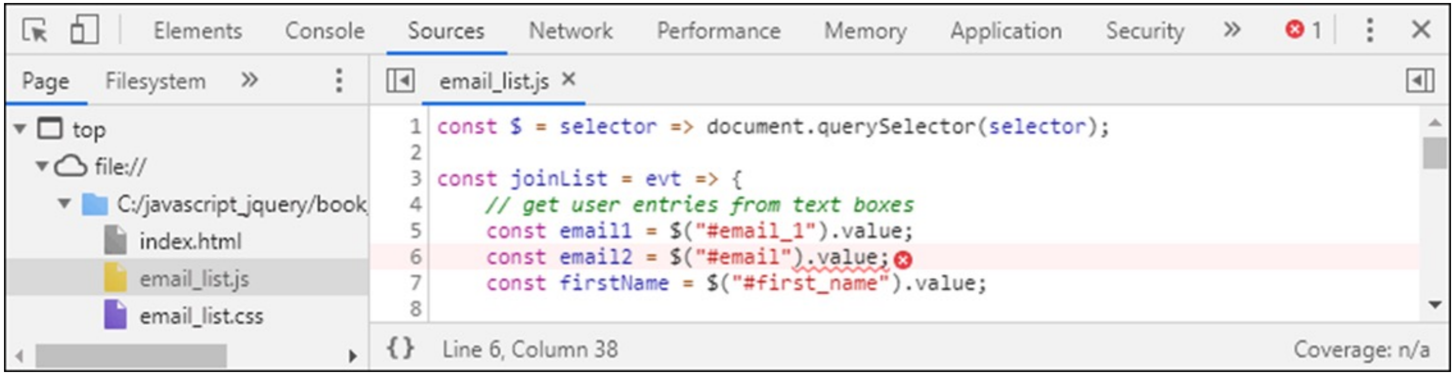
How to open or close Chrome’s developer tools

- To open, press F12 or Ctrl+Shift+I. Or, click on the Menu button in the upper right corner of the browser, and select More Tools→Developer Tools.
- To close, click on the X in the upper right corner of the tools panel or press F12.

How to find the JavaScript statement that caused the error

- Open the Console panel by clicking on the Console tab. You should see an error message along with the line of code that caused the error.
- Click on the link to the right of the error message that indicates the line of code. That will open the Sources panel with the portion of JavaScript code that contains the statement displayed and the statement highlighted.

The Sources panel after the link in the Console panel has been clicked



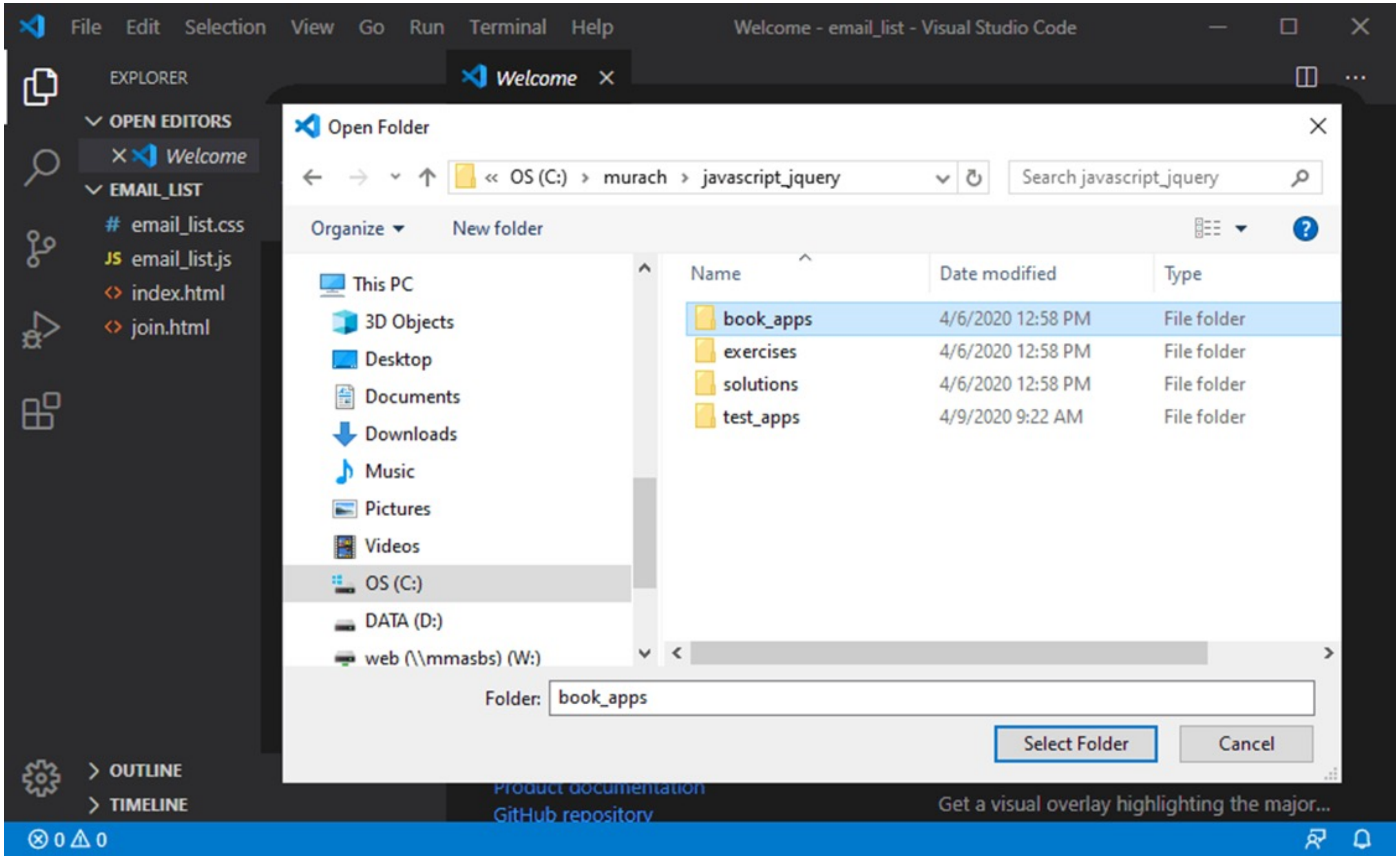
```
const $ = selector => document.querySelector(selector);

const joinList = evt => {
    // get user entries from text boxes
    const email1 = $("#email_1").value;
    const email2 = $("#email").value;
    const firstName = $("#first_name").value;

```

Line 6, Column 38

The dialog box for choosing a folder in VS Code



Name	Date modified	Type
book_apps	4/6/2020 12:58 PM	File folder
exercises	4/6/2020 12:58 PM	File folder
solutions	4/6/2020 12:58 PM	File folder
test_apps	4/9/2020 9:22 AM	File folder

Folder: book_apps

How to open a folder

1. Start VS Code and select File→Open Folder from the menu system.
2. Use the resulting dialog to select the folder that contains the files you want to work with and then click Select Folder.

How to close a folder

- Select File→Close Folder from the menu system.

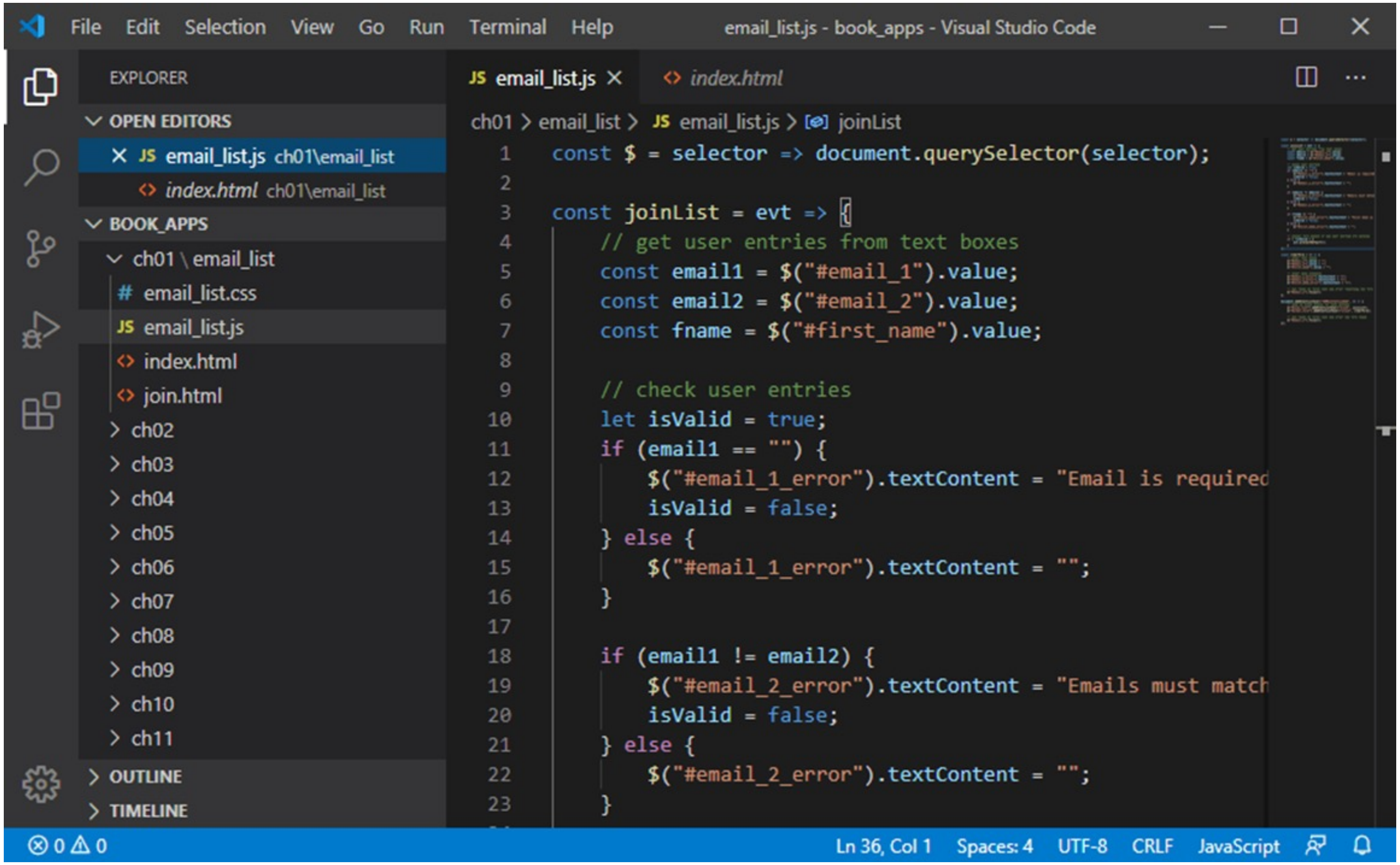
The folder that contains the folders for all of the book applications

```
\murach\javascript_jquery\book_apps
```

How to add, rename, or delete a folder

- To add a folder to the main folder, point to the name of the folder in the Explorer window and click the New Folder icon that's displayed to its right. Then, enter a name for the folder.
- To add other folders, right-click on a folder in the Explorer window and select New Folder. Then, enter a name for the folder.
- To rename a folder, right-click on it and select Rename. Then, edit the name.
- To delete a folder, right-click on it and select Delete.

VS Code with files in Standard and Preview mode



```
const $ = selector => document.querySelector(selector);

const joinList = evt => {
    // get user entries from text boxes
    const email1 = $("#email_1").value;
    const email2 = $("#email_2").value;
    const fname = $("#first_name").value;

    // check user entries
    let isValid = true;
    if (email1 == "") {
        $("#email_1_error").textContent = "Email is required
        isValid = false;
    } else {
        $("#email_1_error").textContent = "";
    }

    if (email1 != email2) {
        $("#email_2_error").textContent = "Emails must match
        isValid = false;
    } else {
        $("#email_2_error").textContent = "";
    }
```

How to preview or open a file

- To open a file, double-click on it in the Explorer window. This displays the file in a tab in the editor with the name of the file in normal font style, indicating that you are in Standard Mode.
- To preview a file, click on it in the Explorer window. This displays the file in a tab in the editor with the name of the file in italics, indicating that you are in Preview Mode. If you open or preview another file, VS Code reuses the tab.
- To display a file that's already open, click on its tab or select it from the Open Editors list at the top of the Explorer window.

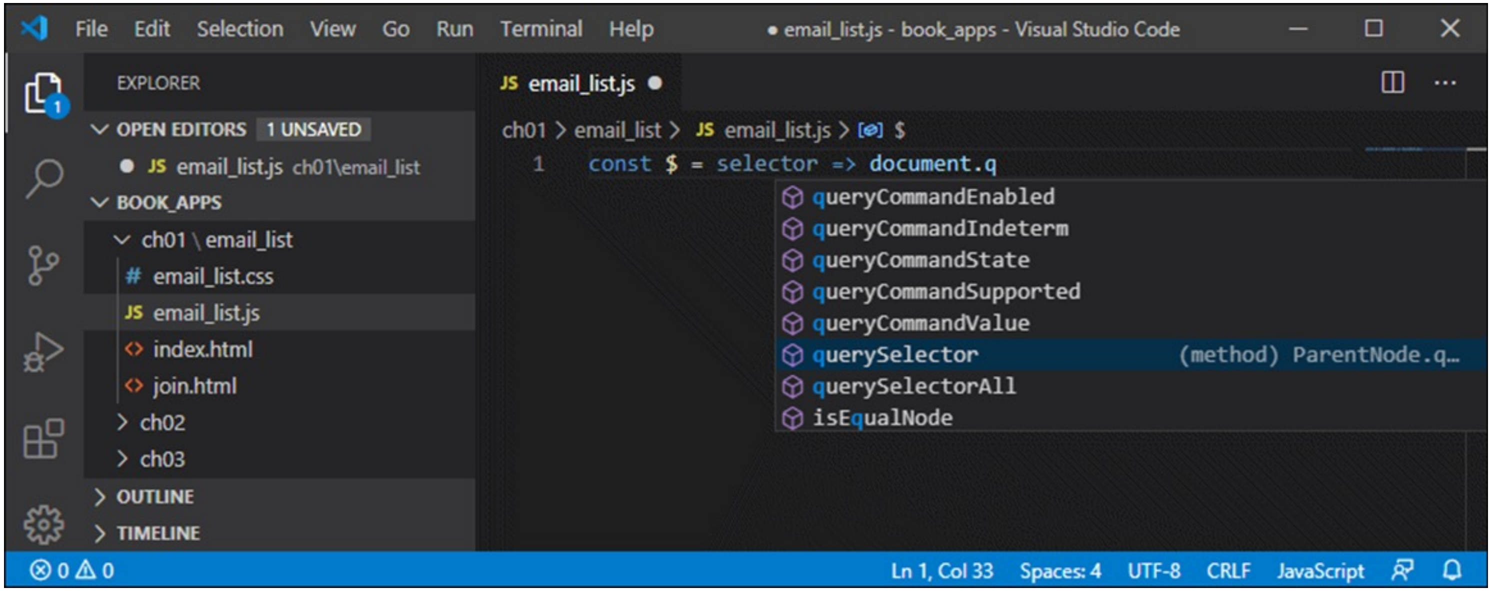
How to close or save a file

- To close a file, click the X in the upper right corner of the tab for the file, click the X to the left of the file name in the Open Editors list, or select File→Close Editor.
- If you close a file with changes, you'll be asked if you want to save the changes.
- If you want to saves changes without closing a file, select File→Save. To save changes to more than one file, select File→Save All.

How to add, rename, or delete a file

- To add a file, you use the same skills as you do for adding a folder except that you click the New File icon or select New File. When you name the file, be sure to include an extension.
- VS Code doesn’t generate any starting code for new files. As a result, you must enter all code for the file yourself or use similar code from another file.
- To rename or delete a file, you use the same skills as you do for renaming or deleting a folder.

The completion list for selecting a property or method of an object



File Edit Selection View Go Run Terminal Help
● email_list.js - book_apps - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
● JS email_list.js ch01\email_list
BOOK_APPS
ch01 \ email_list
# email_list.css
JS email_list.js
<> index.html
<> join.html
> ch02
> ch03
> OUTLINE
> TIMELINE
JS email_list.js ●
ch01 > email_list > JS email_list.js > [ø] $
⊗ 0 ⚠ 0
Ln 1, Col 33 Spaces: 4 UTF-8 CRLF JavaScript

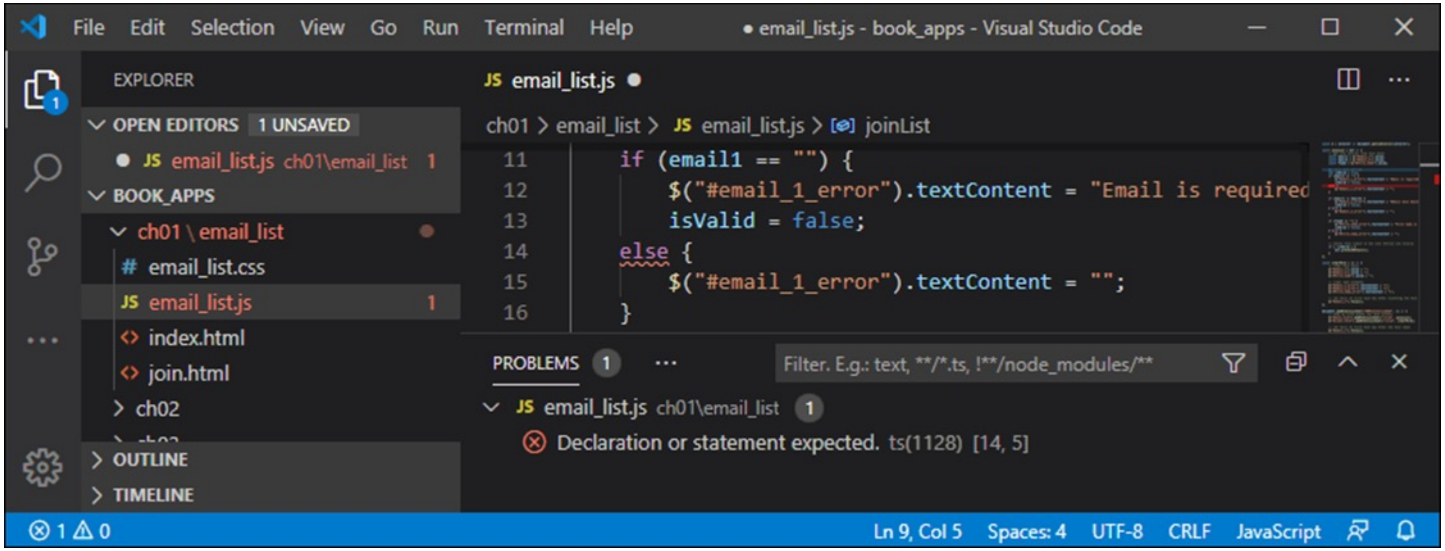
```
const $ = selector => document.q
```

```
queryCommandEnabled
queryCommandIndeterm
queryCommandState
queryCommandSupported
queryCommandValue
querySelector                (method) ParentNode.q…
querySelectorAll
isEqualNode
```

How to use the IntelliSense feature

- IntelliSense displays completion lists for things like keywords, variables, properties, methods, and functions as you type so you can enter them correctly.
- To insert an item from a completion list, click on it or highlight it and then press the Tab or Enter key.
- If you enter an opening parenthesis or brace, the closing parenthesis or brace is added automatically.

The Problems window with an error displayed



```
11        if (email1 == "") {
12            $("#email_1_error").textContent = "Email is required
13            isValid = false;
14        else {
15            $("#email_1_error").textContent = "";
16        }
```

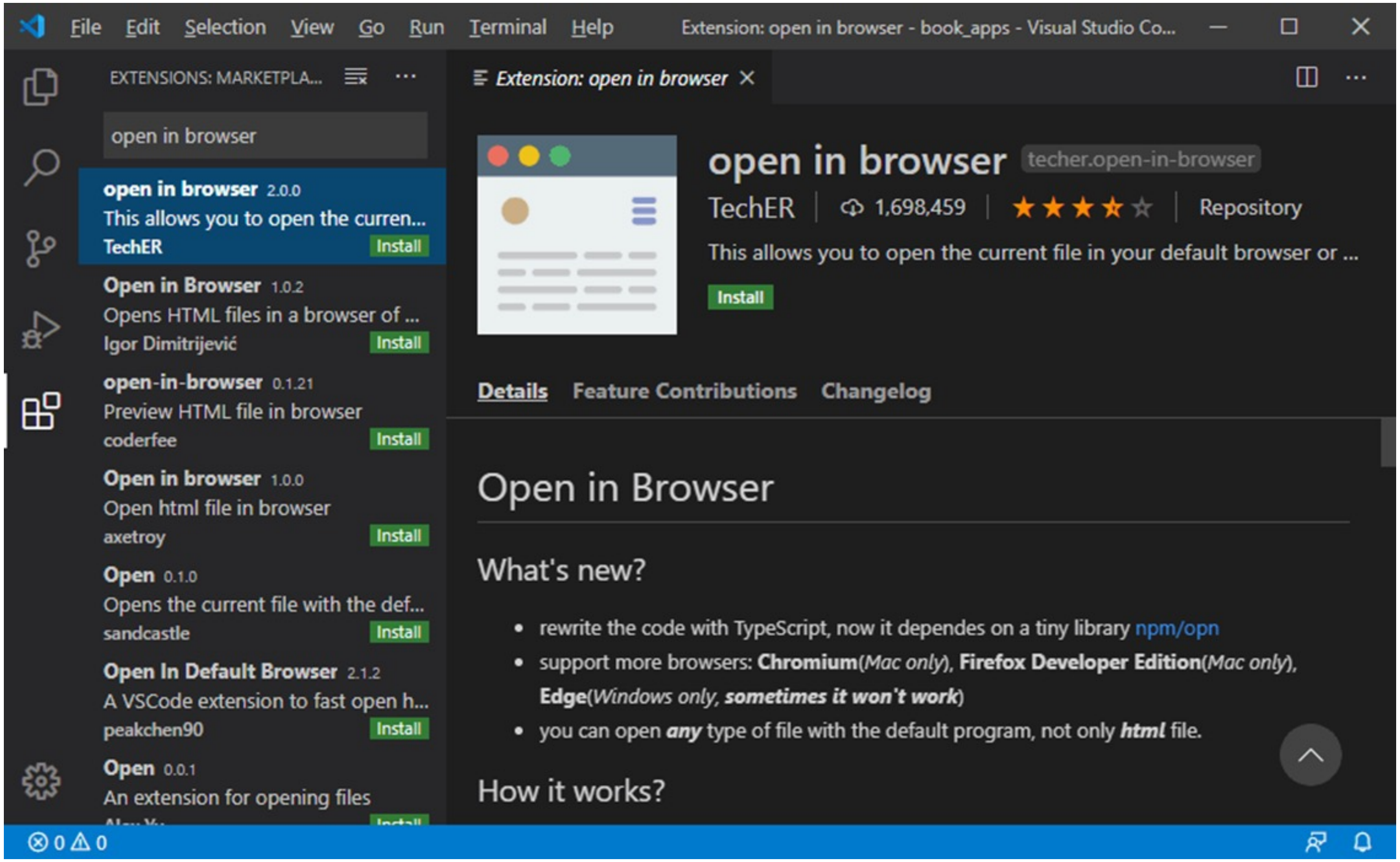
PROBLEMS 1
Filter. E.g.: text, **/*.ts, !**/node_modules/**
JS email_list.js ch01\email_list 1

Declaration or statement expected. ts(1128) [14, 5]

How to identify the errors marked by VS Code

- If VS Code detects a syntax error, it underlines it with a red wavy line.
- To get the description for an error, hover the mouse over the red wavy line.
- To see all the errors in a file, you can display the Problems window (View→Problems). Then, you can click on an error to take you to it in the file.

Installing the Open in Browser extension



open in browser 2.0.0
This allows you to open the curren...
TechER

Open in Browser 1.0.2
Opens HTML files in a browser of ...
Igor Dimitrijević

open-in-browser 0.1.21
Preview HTML file in browser
coderfee

Open in browser 1.0.0
Open html file in browser
axetroy

Open 0.1.0
Opens the current file with the def...
sandcastle

Open In Default Browser 2.1.2
A VSCode extension to fast open h...
peakchen90

Open 0.0.1
An extension for opening files

open in browser techer.open-in-browser
TechER | 1,698,459 | ★★★★☆ | Repository
This allows you to open the current file in your default browser or ...

Open in Browser

What's new?

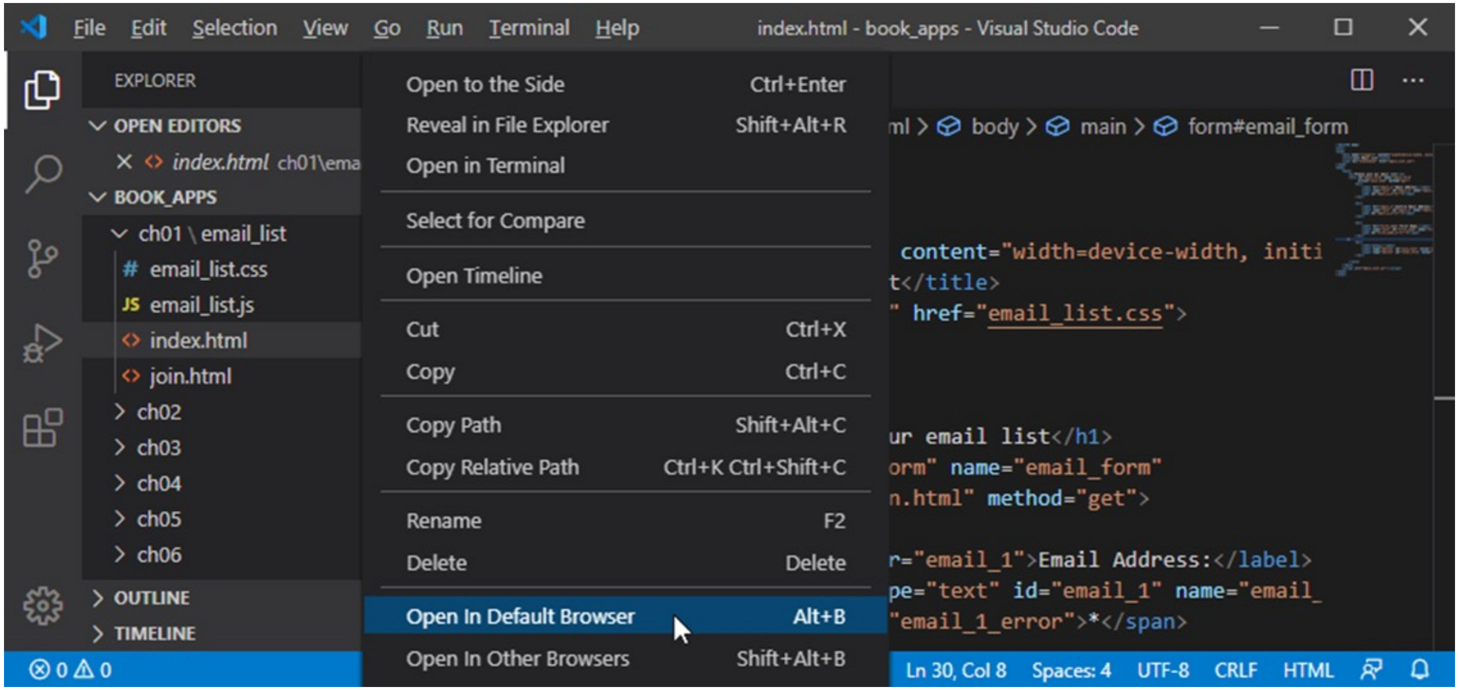
- rewrite the code with TypeScript, now it dependes on a tiny library npm/opn
- support more browsers: **Chromium**(*Mac only*), **Firefox Developer Edition**(*Mac only*), **Edge**(*Windows only, **sometimes it won't work***)
- you can open ***any*** type of file with the default program, not only ***html*** file.

How it works?

How to install the Open in Browser extension

1. Click the Extensions icon in the left sidebar.
2. Enter "open in browser" in the text box at the top of the Extensions window to filter the available extensions.
3. Click the Install button for the Open in Browser extension from TechER. Or, click on the extension to display information about it a tab, and then click the Install button in the tab.

How to open an HTML file using the Open in Browser extension

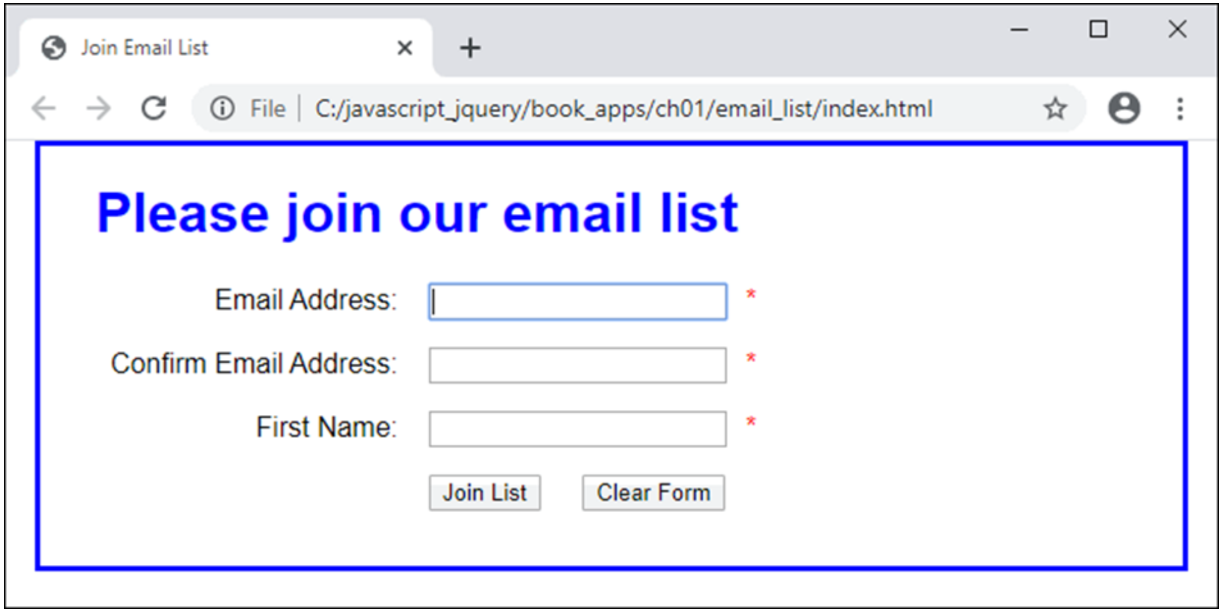


```
 content="width=device-width, initi
t</title>
" href="email_list.css">


ur email list</h1>
orm" name="email_form"
n.html" method="get">

r="email_1">Email Address:</label>
pe="text" id="email_1" name="email_
"email_1_error">*</span>
```

The web page in Chrome



Please join our email list

Email Address: *

Confirm Email Address: *

First Name: *

How to open an HTML file in a browser

- To open an HTML file using the Open in Browser extension, right-click on the file in the Explorer window and select Open in Default Browset to open the file in your default browser.
- To open an HTML file in a browser other than the default, right-click on the file, select Open In Other Browsers, and then select the browser you want to use.
- To open an HTML file without using the Open in Browser extension, right-click on the file and select Reveal in File Explorer (Windows) to display it in File Explorer or Reveal in Finder (macOS) to display it in Finder. Then, you can double-click the file to open it.
- Every time you open an HTML file from VS Code, another browser instance or browser tab is opened. Another alternative is to save the corrected files in VS Code, switch to the browser, and click its Reload or Refresh button.